

Práctica 4. Control de Acceso y SNMP

Administración de Redes

Índice

- Control de Acceso
- Syslog
- SNMP

Control de Acceso

- Hay varios métodos para acceder a la gestión de un router:
 - Puertos de consola
 - Puerto auxiliar
 - Conexiones de terminal virtual
- Además, el acceso al modo privilegiado también debe ser controlado
- En redes grandes se suele emplear un servidor de autenticación centralizado (RADIUS, TACACS+,...) mediante un servidor AAA.
 - Sin embargo, determinadas contraseñas será necesario que se mantengan en los dispositivos
- En redes pequeñas las claves se almacenan en una base de datos mantenida localmente.
- En IOS se emplean niveles de privilegio a la hora de acceder al dispositivo: 0 a 15

Control de Acceso: Configuración

- Las claves se configuran en el modo de configuración global, dentro del modo privilegiado.
 - **enable secret <password>**: Acceso al modo EXEC privilegiado.
 - Clave codificada con algoritmo HASH MD5. A partir de IOS 15.0(1) se usa SHA256
 - Si se pierde la clave hay que emplear un procedimiento de recuperación de claves.
 - Línea de consola.
 - Por defecto no se necesita clave para acceder desde consola.
 - Si se desea que se requiera clave al acceder hay que emplear el comando **line console 0** y luego, en el modo de configuración de consola, activar la solicitud de **login** con la clave correspondiente (**password <password>**)
 - Por defecto esta contraseña se almacena en texto claro

Control de Acceso: Cifrado de Contraseñas

- Por defecto, las claves configuradas mediante el comando **password** se almacenan sin cifrar y por lo tanto pueden ser vistas con **show running-config** o mediante *sniffers* en conexiones TFTP al hacer *backups* de ficheros de configuración.
- El comando de configuración global **service password-encryption** cifra las contraseñas que por defecto se almacenan en texto claro, utilizando el cifrado **Vignère**, que es fácilmente reversible → Solamente debe utilizarse para evitar *shoulder surfing attacks*
- Ejemplo:

```
R1(config)# enable secret class123
R1(config)# line console 0
R1(config-line)# password cisco123
R1(config-line)# login
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# password cisco123
R1(config-line)# login
R1(config-line)# exit
R1(config)# service password-encryption
R1(config)# exit
```

Cifrado de Contraseñas: Verificación

- R1# **show run**
Building configuration...
<Output omitted>
service password-encryption
<Output omitted>
enable secret **4** JpAg4vBxn6wTb6NE3N1p0wfUUZzR6eOcVUKUfftxEyA
<Output omitted>
line con 0
 password **7** 070C285F4D06485744
 login
line aux 0
line vty 0 4
 password **7** 070C285F4D06485744
 login
 transport input all
!
end
- Códigos de cifrado: Tipo 0 → Texto Claro; Tipo 4 → Hash SHA256; Tipo 5 → Hash MD5; Tipo 7 → Cifrado Vignère

Usuarios locales

- El cifrado de contraseña de línea utilizado para el acceso a través de consola y líneas VTY es muy débil.
- Se recomienda emplear una base de datos local que contenga la lista de usuarios y claves. Los usuarios se pueden crear mediante dos comandos:
 - `username <name> password <password>`
 - `username <name> secret <password>`
- La segunda opción es más segura puesto que emplea el algoritmo MD5 o SHA256, que son mucho más seguros que Vignere, que es el que emplea el comando `service password encryption`
- Para habilitar la base de datos local para autenticación se puede emplear el comando `login local`

Usuarios Locales

- Ejemplo de configuración

```
R1(config)# username ADMIN secret class12345
R1(config)# username JR-ADMIN secret class123
R1(config)# line console 0
R1(config-line)# login local
R1(config-line)# exit
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# end
```

- Verificación de la autenticación local

```
R1# show running-config | include username
username ADMIN secret 4 VYlArd0J6s2X4dZwZ42oTpLQ5Zog8wZDgZKHMP2SHEw
username JR-ADMIN secret 4
JpAg4vBxn6wTb6NE3N1p0wfUUZzR6eOcVUKUfftxEyA
R1#exit
R1 con0 is now available
Press RETURN to get started.
User Access Verification
Username: ADMIN
Password:
R1>
```


Autenticación, Autorización y Auditoría (AAA)

- Una red corporativa debe estar diseñada para controlar “quién” se conecta (**autenticación**) y “qué” puede hacer cuando se conecta (**autorización**)
- También es necesario implementar sistemas de contabilización o **auditoría** (accounting) que permitan hacer un seguimiento sobre cuándo y qué han hecho los usuarios conectados
- AAA es un *framework* basado en estándares que se utiliza para controlar el acceso de gestión a los dispositivos de red implementando mecanismos de autenticación, autorización y auditoría
 - Base de datos de usuario local
 - Base de datos de usuarios ubicada en un servidor externo, por ejemplo un Directorio Activo de Microsoft o OpenLDAP.
- Ventajas:
 - Incrementa la flexibilidad y el control de acceso a la configuración
 - Es escalable
 - Permite el uso de diferentes métodos de *backup*
 - Utiliza métodos de autenticación estandarizados

Autenticación, Autorización y Auditoría (AAA)

- Los usuarios deben autenticarse contra una BD. Hay dos opciones de **autenticación AAA**:
 - **Local AAA**: En este caso hay una base de datos local.
 - La BD es la misma que la que se emplea para roles en el router → Método ideal para redes pequeñas
 - Las implementaciones de AAA locales no son fácilmente escalables a redes grandes.
 - La opción de trabajar con AAA basado en servidor soluciona este problema
 - Basado en **servidor AAA**: Se emplea un Server BD externo
 - Se emplean protocolos de comunicación seguros como RADIUS.
 - Un ejemplo de esta implementación es Cisco ACS, pero existen otras como FreeRADIUS

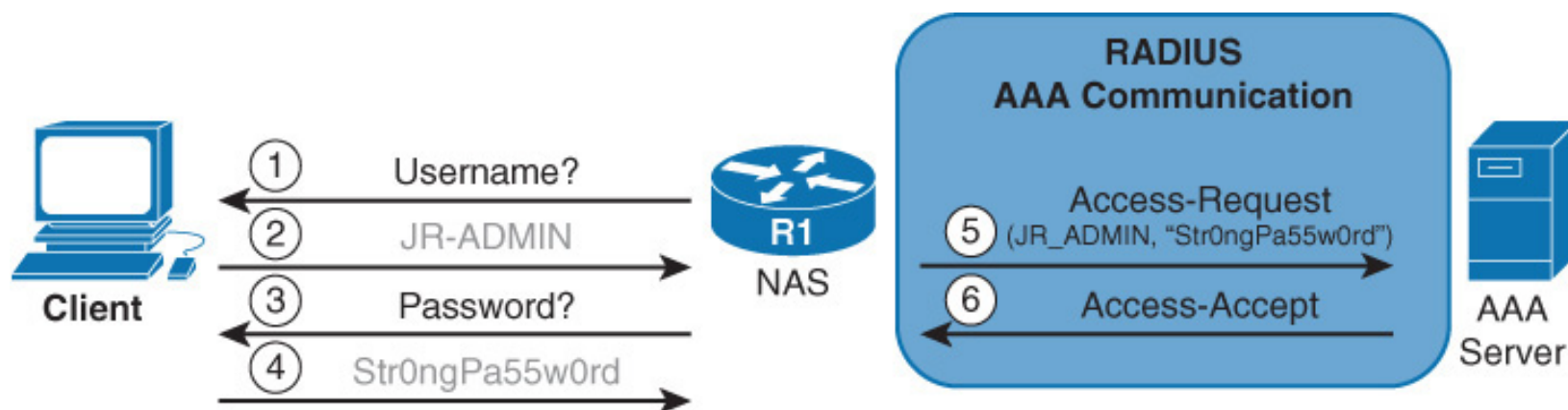
Características de AAA basada en servidor

- **RADIUS: Remote Dial-in User Services**

- Permite comunicar el router (o switch) y el servidor AAA
- RADIUS sólo cifra la clave del usuario:
 - Oculta la clave utilizando un mecanismo derivado de MD5 y una clave secreta
 - El nombre de usuario y el resto de información se transmite en texto claro
- Es un estándar IETF
- Se utiliza mucho en los ISP porque es capaz de gestionar información detallada de facturación (por ejemplo gestión de pagos)
- Los servidores Proxy emplean RADIUS por su escalabilidad
- Utiliza UDP
- La autenticación y autorización van unidas en un único proceso
 - Puertos UDP 1645 o 1812 para la autenticación
 - Puertos UDP 1646 o 1813 para “accounting”.
- Da servicio a tecnologías de acceso remoto, 802.1x y SIP
- Se considera que el protocolo DIAMETER reemplazara a RADIUS empleando SCTP (Stream Control Transmission Protocol) y TCP

Características de AAA basada en servidor

- Proceso de Autenticación basado en un Servidor RADIUS



Configuración de autenticación local AAA

1. Crear una base de datos de usuarios locales mediante `username <name> secret <password>`
2. Habilitar AAA globalmente en el router se emplea el comando `aaa new-model`
3. A continuación es necesario definir una **lista de métodos de autenticación** y aplicarla a las vías de acceso al dispositivo.
 - Para definir esta lista de métodos de autenticación se usa el comando:
`aaa authentication login {default|list_name} metodo_1 ...metodo_4`
 - La lista identifica diferentes protocolos de seguridad para autenticar al usuario.
 - Pueden utilizarse varios métodos por seguridad para garantizar alternativas en caso de que el método anterior no este disponible.
 - Ejemplo:

```
aaa authentication login TELNET-ACCESS group radius enable
```

En este caso primero se intenta autenticar al usuario mediante la base de datos de un servidor RADIUS y si este método falla (e.g. servidor caído) mediante la clave de acceso a modo privilegiado

Configuración de autenticación local AAA

4. A continuación, se aplican las listas de métodos de autenticación a interfaces.
- Es posible crear una **lista por defecto** que se aplica automáticamente a todos los interfaces y líneas mediante `aaa authentication local default metodo1...metodo4` siempre y cuando no haya una lista específica creada para un interfaz concreto, en cuyo caso esa lista para ese interfaz sobrescribe las configuraciones de la lista por defecto.

```
R1(config)# username JR-ADMIN secret Str0ngPa55w0rd
R1(config)# username ADMIN secret Str0ng5rPa55w0rd
R1(config)#
R1(config)# aaa new-model
R1(config)# aaa authentication login default local
R1(config)# aaa authentication login TELNET-LOGIN local-case
R1(config)# line vty 0 4
R1(config-line)# login authentication TELNET-LOGIN
R1(config-line)# exit
```

Configuración de autenticación local AAA

- Se puede ampliar la seguridad con el comando `aaa local authentication attempts max-fail [number_of_unsuccessful_attempts]` de modo que se bloquean las cuentas tras un determinado nº de intentos de autenticación incorrectos
 - El comando `show aaa local user lockout` permite ver los usuarios actualmente bloqueados.
 - Para borrar la lista de usuarios bloqueados se puede emplear el comando `clear aaa local user lockout {username username | all}`
- También hay un comando `login delay` que introduce un retardo entre intentos de autenticación sin bloquear la cuenta.
- Para mostrar información acerca de una sesión de un usuario podemos emplear el comando `show aaa user {all | unique_id}` el cual proporciona información de los usuarios autenticados con AAA.
- El comando `show aaa sessions` muestra el `unique_id` de una sesión.

Troubleshooting Local AAA Authentication

- El comando fundamental de localización de errores es **debug aaa** con multitud de variantes entre ellas `debug aaa authentication`
- Es importante prestar atención a los mensajes de estado **GETUSER** y **GETPASS**
- Para deshabilitar el comando emplear “no ...” o bien “undebug all”

```
R1# debug aaa authentication
113123: Feb 4 10:11:19.305 CST: AAA/MEMORY: create_user (0x619C4940) user=''
ruser='' port='ttyl' rem_addr='async/81560' authen_type=ASCII service=LOGIN priv=1
113124: Feb 4 10:11:19.305 CST: AAA/AUTHEN/START (2784097690): port='ttyl' list=''
action=LOGIN service=LOGIN
113125: Feb 4 10:11:19.305 CST: AAA/AUTHEN/START (2784097690): using "default" list
113126: Feb 4 10:11:19.305 CST: AAA/AUTHEN/START (2784097690): Method=LOCAL
113127: Feb 4 10:11:19.305 CST: AAA/AUTHEN (2784097690): status = GETUSER
113128: Feb 4 10:11:26.305 CST: AAA/AUTHEN/CONT (2784097690): continue_login
(user='(undef)')
113129: Feb 4 10:11:26.305 CST: AAA/AUTHEN (2784097690): status = GETUSER
113130: Feb 4 10:11:26.305 CST: AAA/AUTHEN/CONT (2784097690): Method=LOCAL
113131: Feb 4 10:11:26.305 CST: AAA/AUTHEN (2784097690): status = GETPASS
113132: Feb 4 10:11:28.145 CST: AAA/AUTHEN/CONT (2784097690): continue_login
(user='diallocal')
113133: Feb 4 10:11:28.145 CST: AAA/AUTHEN (2784097690): status = GETPASS
113134: Feb 4 10:11:28.145 CST: AAA/AUTHEN/CONT (2784097690): Method=LOCAL
113135: Feb 4 10:11:28.145 CST: AAA/AUTHEN (2784097690): status = PASS
```


Autenticación AAA basada en servidor

Paso 1: Habilitar globalmente AAA para permitir el uso de todos los elementos AAA.

Paso 2: Especificar el servidor que proporcionara servicios AAA al router.

Paso 3: Configurar la clave de cifrado necesaria para cifrar los datos entre el servidor de acceso de la red y AAA

Paso 4: Configurar la lista de métodos de autenticación AAA.

Autenticación AAA basada en servidor

- **Configuración de un servidor RADIUS y la clave de cifrado:**
 - Especificación del servidor AAA accesible mediante RADIUS `radius-server host <ip_address>`
 - Especificamos la clave de cifrado en el router con el comando: `radius-server key`
 - Esta misma clave ha de ser configurada en el servidor RADIUS
- **Configuración de la Autenticación para usar un servidor AAA:**
 - `aaa authentication login`

Autenticación AAA basada en servidor

```
R1(config)# username JR-ADMIN secret Str0ngPa55w0rd
R1(config)# username ADMIN secret Str0ng5rPa55w0rd
R1(config)# aaa new-model
R1(config)# radius server RADIUS-1
R1(config-radius-server)# address ipv4 192.168.1.101
R1(config-radius-server)# key RADIUS-1-pa55w0rd
R1(config-radius-server)# exit
R1(config)# radius server RADIUS-2
R1(config-radius-server)# address ipv4 192.168.1.102
R1(config-radius-server)# key RADIUS-2-pa55w0rd
R1(config-radius-server)# exit
R1(config)# aaa group server radius RADIUS-GROUP
R1(config-sg-radius)# server name RADIUS-1
R1(config-sg-radius)# server name RADIUS-2
R1(config-sg-radius)# exit
R1(config)# aaa authentication login default group RADIUS-GROUP local
R1(config)# aaa authentication login TELNET-LOGIN group RADIUS-GROUP local-case
R1(config)# line vty 0 4
R1(config-line)# login authentication TELNET-LOGIN
R1(config-line)# exit
R1(config)#
```

Autenticación AAA basada en servidor

- Podemos emplear el comando **debug aaa authentication** que muestra mucha información durante el proceso de autenticación.

```
R1# debug aaa authentication
AAA Authentication debugging is on
R1#
14:01:17: AAA/AUTHEN (567936829): Method=TACACS+
14:01:17: TAC+: send AUTHEN/CONT packet
14:01:17: TAC+ (567936829): received authen response status = PASS
14:01:17: AAA/AUTHEN (567936829): status = PASS
```

- Otros comandos son **debug radius** que muestran información muy detallada.
- Podemos emplear el comando **debug aaa authentication** para mostrar información acerca de los mensajes de estado de la autenticación correcta o con fallo.

Configurando Autorización AAA basada en servidor

- La autenticación busca asegurar el acceso al dispositivo
- La autorización se refiere a permitir y no permitir a los usuarios autenticados acceso a ciertas áreas y programas en la red.
- El protocolo RADIUS no separa la autenticación del proceso de autorización.
- Mediante mecanismos de autorización se puede permitir o no la ejecución de determinados comandos, puesto que el router consulta al servidor AAA para saber si el usuario tiene permiso para ejecutar el comando concreto.

Configurando Autorización AAA basada en servidor

- Para configurar la autorización emplearemos el comando `aaa authorization {network | exec | commands level} {default | list-name} method1...[method4] ```
 - **Commands level:** para comandos EXEC
 - **Exec:** para iniciar un exec (shell)
 - **Network:** para servicios de red (PPP, SLIP...)
- Cuando se inicia la autenticación ya se deben haber creado los usuarios con derechos de acceso si no se bloquearía al administrador y sería necesario reiniciar.

```
R1(config)# aaa authorization exec ?  
WORD      Named authorization list.  
default   The default authorization list.  
  
R1(config)# aaa authorization exec default
```

```
R1(config)# aaa authorization exec default ?  
group      Use server-group.  
if-authenticated Succeed if user has authenticated.  
krb5-instance Use Kerberos instance privilege maps.  
local      Use local database.  
none       No authorization (always succeeds).  
  
R1(config)# aaa authorization exec default group ?  
WORD      Server-group name  
radius     Use list of all Radius hosts.  
tacacs+    Use list of all Tacacs+ hosts.
```

```
R1# conf t  
R1(config)# username JR-ADMIN secret Str0ngPa55w0rd  
R1(config)# username ADMIN secret Str0ng5rPa55w0rd  
R1(config)# aaa new-model  
R1(config)# aaa authentication login default group tacacs+  
R1(config)# aaa authorization exec default group tacacs+  
R1(config)# aaa authorization network default group tacacs+
```

Configurando Auditoría AAA basada en servidor

- A veces una empresa quiere hacer un seguimiento de los recursos que los individuos o grupos usan.
- El servicio *Accounting* AAA ofrece la posibilidad de rastrear el uso de los recursos, recoger información en una BD y producir informes sobre los datos recogidos.
 - Ejemplo:
 - Creación de listas de usuarios y hora y día de acceso al sistema
 - Lista de cambios que se producen en la red, quién realizó dichos cambios y la naturaleza exacta de los cambios.
- El servidor AAA actúa como un repositorio central de información “contable” haciendo un seguimiento de los eventos que ocurren en la red.
- Cada sesión que se establece a través del servidor AAA es monitorizada y la información relacionada almacenada, lo cual puede ser muy útil para la gestión y las auditorías de seguridad.
- Es necesario definir listas de métodos de *accounting* que definen el modo en que se realiza la gestión de la contabilidad.

Configurando SSH

- Tradicionalmente las tareas de configuración remota se hacían a través del protocolo telnet que envía la información en texto claro.
- SSH frente a telnet proporciona confidencialidad e integridad en las comunicaciones. La funcionalidad es similar a Telnet, en consecuencia de este modo podemos obtener comunicaciones seguras a través de redes no seguras.
- Los pasos para configurar SSH son las siguientes:
 - Asegurar que los routers disponen de una imagen IOS que soporte SSH lo cual ocurre a partir de la versión 12.1(1)T.
 - Asegurar que cada router tiene un nombre de host único.
 - Asegurar que cada router tiene un nombre de dominio correcto.
 - Asegurar de que los routers están configurados para autenticación local AAA.

Configurando SSH

- PASO 1: El router debe tener un nombre de host único, pero también hay que configurar el nombre de dominio con el comando `ip domain-name <domain-name>`
- PASO 2: Generar un par de claves pública/privada asociado con una dirección y nombre
 - Cisco IOS emplea el algoritmo de claves asimétricas RSA
 - El comando a emplear es `crypto key generate rsa general-keys modulus <modulus-size>`
 - La clave RSA puede ser de 360 a 2048 bits (Mínimo recomendado de 1024 bits)
 - Para verificar el funcionamiento emplearemos el comando `show crypto key mypubkey rsa`
 - Si ya existiera una pareja de claves podemos usar el siguiente comando para sobrescribir las claves `crypto key zeroize rsa`
- PASO 3: Asegurar que el usuario existe en la base de datos local, en caso contrario se usa el comando `username <name> secret <secret>`
- PASO 4: Habilitar las conexiones entrantes VTY SSH con el comando `login local` y `transport input ssh`

Configurando SSH

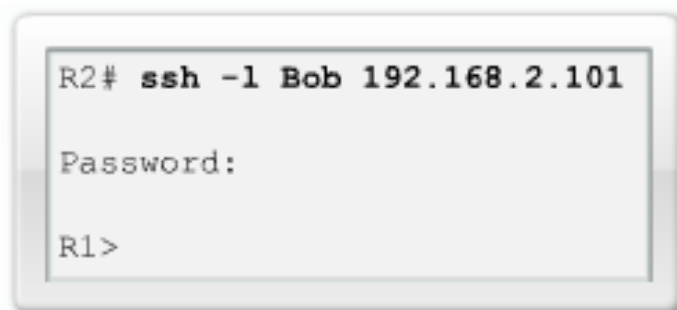
- Comandos opcionales SSH
 - Versión SSH
 - Periodo de Timeout SSH
 - Número de intentos de autenticación.
- Hay dos versiones soportadas SSHv1 y SSHv2 que incorpora el algoritmo Diffie-Hellman como mecanismo de intercambio de claves e incorpora mejoras relacionadas con la integridad a través de MAC (Message Authentication Code)
 - Cisco IOS 12.1(1)T y posteriores soportan SSHv1.
 - Cisco IOS 12.3(4)T y posteriores soportan SSHv1 y SSHv2.
 - El comando que lo permite es `ip ssh version {1|2}`
- Para controlar el tiempo que el router espera a una respuesta por parte del cliente mediante el comando `ip ssh time-out seconds`. Por defecto es 120 segundos
- Para controlar el número de intentos antes de que un usuario sea desconectado podemos emplear el comando `ip ssh authentication-retries integer`
- Para verificar las configuraciones anteriores tenemos disponible el comando `show ip ssh`

Configurando SSH

- Utilización de ACLs para el acceso de gestión al router:

```
R1(config)# ip access-list standard PERMIT-SSH
R1(config-std-nacl)# remark ACL permitting SSH to hosts on the Management LAN
R1(config-std-nacl)# permit 10.0.0.0 0.0.0.255
R1(config-std-nacl)# deny any log
R1(config-std-nacl)# exit

R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# access-class PERMIT-SSH in
R1(config-line)# end
R1#
```



- Cómo conectarse a un a un router con SSH habilitado
 - Conexión empleando el comando “ssh” (Conexión Router-Router)
 - Conectar a través de un cliente SSH (Por ejemplo Putty o OpenSSH) (Conexión Host-Router)

Network Time Protocols

Introducción

- Mantener una configuración de tiempo consistente en todos los equipos de una red es un elemento clave para aspectos relacionados con seguridad o resolución de problemas:
 - Mediciones de SLA
 - Información de syslog
 - Validez de certificados
 - Lanzamiento de *scripts*
- Desde el punto de vista de un dispositivo de red, el reloj de sistema arranca en el momento que el propio sistema se inicia y mantiene actualizados fecha y hora
- El reloj de sistema puede ser configurado desde diferentes orígenes y puede utilizarse, a su vez, para configurar la hora en otros dispositivos
- Internamente, el reloj de sistema marca la hora en base a UTC (*Coordinated Universal Time*)
- Todos los dispositivos Cisco soporta configuración de zona horaria y cambio de hora automático, para mostrar la hora en un formato correcto localmente

Introducción

- El reloj de sistema también puede saber si la hora se ha aprendido desde una fuente autoritativa
 - Si no lo es, la hora solo se utiliza para mostrarse. No se distribuye
 - Autoritativa → La fuente de información de la hora es fiable
- Buenas prácticas:
 - Configurar la hora en formato UTC en todos los dispositivos, independientemente de su ubicación
 - Configurar la zona horaria, para visualizar la hora local

Configuración Manual del Reloj de Sistema

- Si se está ejecutando NTP, el calendario se actualiza periódicamente desde NTP, compensando la desviación horaria que pueda existir.
- Si el sistema no tiene batería, la hora inicial del sistema es constante, dependiendo de la fecha de fabricación del mismo
- En Cisco, el término **calendar** se utiliza para hacer referencia al reloj hardware. Para configurarlo se utiliza el comando

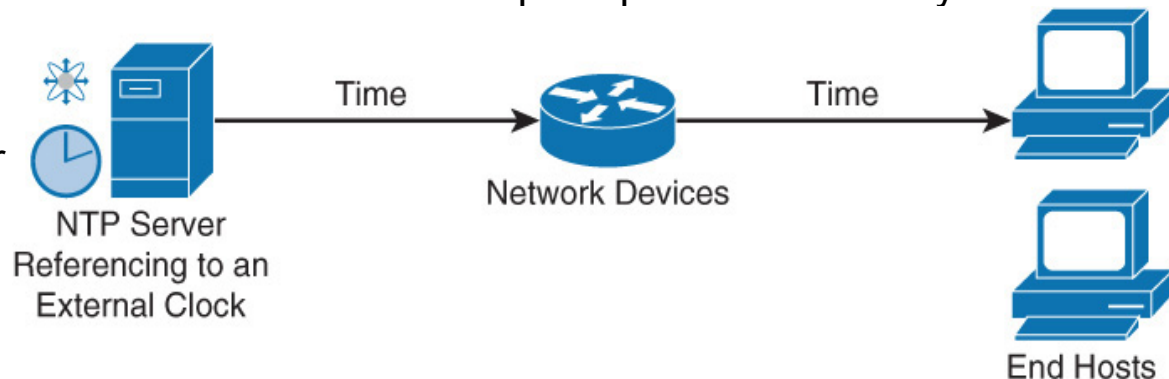
```
calendar set hh:mm:ss <1:31> month year
```

- En caso de no disponer de NTP, se recomienda configurar manualmente **calendar**
- Sin embargo, la configuración manual no es escalable ni precisa. Las buenas prácticas recomiendan utilizar NTP, SNTP o PTP.

Network Time Protocol

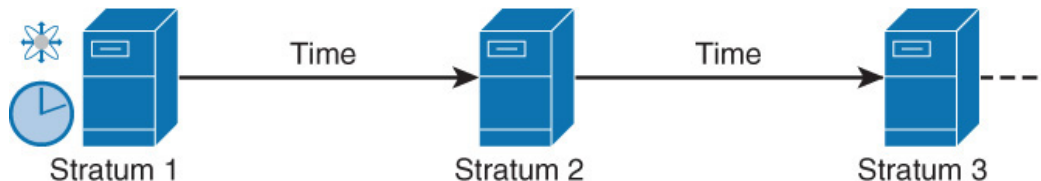
- NTP es un protocolo de capa de aplicación diseñado para sincronizar la hora en una infraestructura de red completa
 - Es utilizado por todo tipo de dispositivos finales e intermedios
 - Encapsula sus mensajes UDP y utiliza el puerto 123 tanto en origen como en destino
 - Sigue el paradigma cliente – servidor
 - Pueden existir múltiples **masters** (servidores principales), pero no es un requisito
- Ejemplo: Un reloj atómico genera la señal de reloj, GPS y otro tipo de información orientada a facilitar la precisión de la hora y envía la información a los dispositivos de red utilizando NTP
- Arquitectura NTP: Existe una **fuentes de información de hora autoritativa** (GPS, reloj atómico, reloj de radio, etc.) que se conecta a un **servidor** principal. NTP distribuye esta información por toda la red.

- Los clientes NTP hacen transacciones con el server periódicamente (64 a 1024 segundos)
 - Ajuste dinámico



Network Time Protocol

- Las asociaciones NTP se configuran, habitualmente, de forma estática
 - Cada dispositivo tiene la IP de los otros dispositivos con los que tiene que asociarse
 - Se mantiene la precisión de la hora intercambiando mensajes NTP entre cada par de máquinas asociadas
- Sin embargo, en una LAN, NTP puede configurarse para enviar mensajes *broadcast* IP
 - Reduce la complejidad de la configuración
 - La precisión puede reducirse porque los mensajes se transmiten en un solo sentido
- NTP utiliza el concepto **stratum** para indicar el número de saltos que hay desde una máquina a la fuente de información de hora autoritativa
 - Stratum 1 → Directamente conectado a la fuente autoritativa, que tendría nivel de stratum 0
 - Una máquina cliente NTP elige siempre la información procedente del stratum menor
 - Esto facilita la creación de árboles auto-organizativos de *NTP speakers*



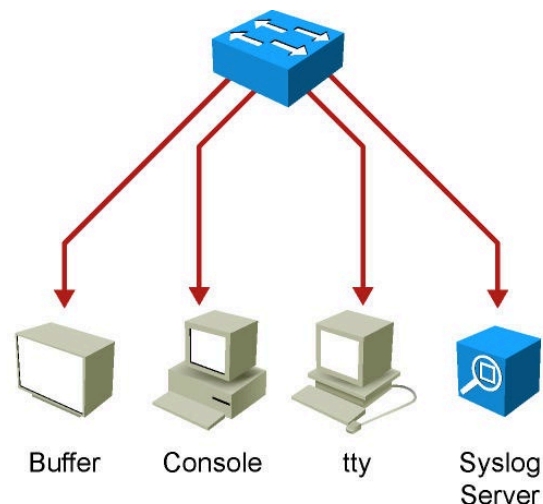
Network Time Protocol

- NTP funciona bien sobre estructuras con rutas no determinísticas de redes de conmutación de paquetes porque hace estimaciones robustas basadas en la relación entre el cliente y el servidor y tres variables:
 - Retardo de red
 - Dispersión de tiempo en el intercambio de paquetes o lo que es lo mismo una medida del máximo error de reloj entre 2 hosts
 - *Clock offset* o desplazamiento de reloj: Corrección aplicada al reloj cliente para sincronizarlo
- Se puede conseguir una sincronización de:
 - 10 ms sobre WAN de larga distancia (2.000 Km)
 - 1 ms sobre LANs
- NTP evita de dos formas la sincronización con una máquina cuya hora no sea precisa:
 - Nunca se sincroniza con una máquina que no está sincronizada consigo misma
 - Compara la información recibida desde varias fuentes y no se sincroniza con aquella que tenga una diferencia significativa con las demás

Syslog

Syslog

- System Message Logging es un proceso que **permite a un dispositivo informar de mensajes de error y notificación importantes**
 - Local o remoto: Se envían por defecto a la consola, aunque pueden redirigirse (buffer local o servidor remoto)
 - **Usa el puerto UDP 514.**
 - Registra los mensajes del sistema en texto plano (inglés).
- Los mensajes se generan durante el funcionamiento de la red para ayudar a **identificar el tipo y la gravedad de un problema** o para ayudar a los administradores a supervisar la actividad del router cuando se producen **cambios en la configuración.**
- Los comandos **debug** muestran información a través de mensajes de Syslog
- Todos los mensajes de syslog contienen **un nivel de gravedad y una causa o programa causante (facility)**
- Routers, switches, servidores, firewalls u otros tipos de dispositivos de red soportan generan mensajes de **Syslog**



Niveles de Severidad de Syslog

Syslog Severity	Severity Level
Emergency	Level 0, highest level
Alert	Level 1
Critical	Level 2
Error	Level 3
Warning	Level 4
Notice	Level 5
Informational	Level 6
Debugging	Level 7


Syslog Facilities

- Son identificadores de servicio que permiten especificar y clasificar los datos de estado del sistema para la notificación de mensajes de evento y error.
 - Cisco IOS tiene más de 500 identificadores o «facilities».
 - Las “syslog facilities” más comunes:
 - IP
 - OSPF
 - SYS operating system
 - IP Security (IPsec)
 - Route Switch Processor (RSP)
 - Interface (IF)
 -
 - CDP
 - STP
 -

Formato de Mensajes de Syslog

- Los mensajes del sistema comienzan con un signo de porcentaje (%)
- **Facility:** Código formado por dos o más letras mayúsculas que indica el dispositivo hardware, protocolo o módulo de software del sistema
- **Severity:** Dígito (0 a 7) que refleja la gravedad. El nº menor → Mayor gravedad
- **Mnemonic:** Código que identifica no ambigua el mensaje de error.
- **Message-text:** Cadena de texto que describe la condición. Puede contener información detallada sobre el evento (números de puerto terminal, direcciones de red ...)
- Los mensajes se muestran en este formato:
 - **seq no:timestamp: %facility-severity-MNEMONIC:description**
 - **El número de secuencia y la marca horaria aparecen si están configurados en los dispositivos.**

%FACILITY-SUBFACILITY-SEVERITY-MNEMONIC: Message-text



```
%SYS-5-CONFIG_I: Configured from console by
cwr2000 on vty0 (192.168.64.25)
```

Ejemplo de Mensajes de Syslog

```
08:01:13: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up
08:01:23: %DUAL-5-NBRCHANGE: EIGRP-IPv4:(1) 1: Neighbor 10.1.1.1 (Vlan1) is up: new adjacency
08:02:31: %LINK-3-UPDOWN: Interface FastEthernet0/8, changed state to up
08:18:20: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to
down
08:18:22: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/5, changed state to up
08:18:24: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to
down
08:18:24: %ILPOWER-5-IEEE_DISCONNECT: Interface Fa0/2: PD removed
08:18:26: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to down
08:19:49: %ILPOWER-7-DETECT: Interface Fa0/2: Power Device detected: Cisco PD
08:19:53: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
08:19:53: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed state to up
```

- Los mensajes más comunes son los de activación/desactivación de las interfaces.
- Si el logging basado en ACLs está configurado, el dispositivo genera un mensaje cuando los paquetes coinciden con la condición definida en la ACL.

Configuración de Syslog (1)

- Para configurar cual es el servidor de syslog al que se envían los mensajes, se utiliza el comando de configuración global **logging** *ip_addr*.
- Para determinar qué niveles de gravedad de los mensajes se envían al servidor syslog, se utiliza el comando de configuración global **logging trap** *level*

```
Switch(config)# logging trap ?
```

<0-7>	Logging severity level	
alerts	Immediate action needed	(severity=1)
critical	Critical conditions	(severity=2)
debugging	Debugging messages	(severity=7)
emergencies	System is unusable	(severity=0)
errors	Error conditions	(severity=3)
informational	Informational messages	(severity=6)
notifications	Normal but significant conditions	(severity=5)
warnings	Warning conditions	(severity=4)

Configuración de Syslog (2)

- Para configurar el registro para el búfer del switch local, utilizamos el comando **logging buffered**.

```
Switch(config)# logging buffered ?
```

<0-7>	Logging severity level	
<4096-2147483647>	Logging buffer size	
alerts	Immediate action needed	(severity=1)
critical	Critical conditions	(severity=2)
debugging	Debugging messages	(severity=7)
discriminator	Establish MD-Buffer association	
emergencies	System is unusable	(severity=0)
errors	Error conditions	(severity=3)
informational	Informational messages	(severity=6)
notifications	Normal but significant conditions	(severity=5)
warnings	Warning conditions	(severity=4)
xml	Enable logging in XML to XML logging buffer	

Comprobación de la Configuración de Syslog

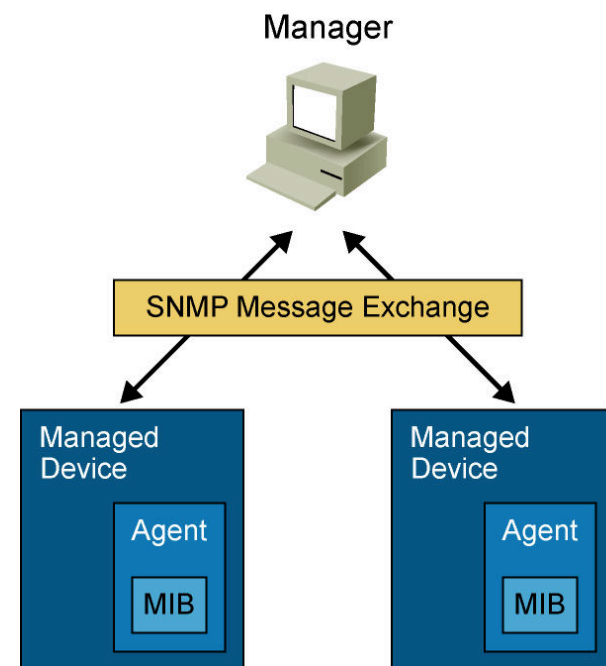
- **show logging** → mostrar el contenido de los archivos de log locales.
- El argumento pipe (|) en combinación con palabras clave tales como **include** ó **begin** para filtrar la salida.

```
Switch# show logging | include LINK-3
2d20h: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
2d20h: %LINK-3-UPDOWN: Interface FastEthernet0/2, changed state to up
2d20h: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
Switch# show logging | begin %DUAL
2d22h: %DUAL-5-NBRCHANGE: EIGRP-IPv4:(10) 10: Neighbor 10.1.253.13
(FastEthernet0/11) is down: interface down
2d22h: %LINK-3-UPDOWN: Interface FastEthernet0/11, changed state to down
2d22h: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/11,
changed state to down
```

SNMP

Introducción a SNMP

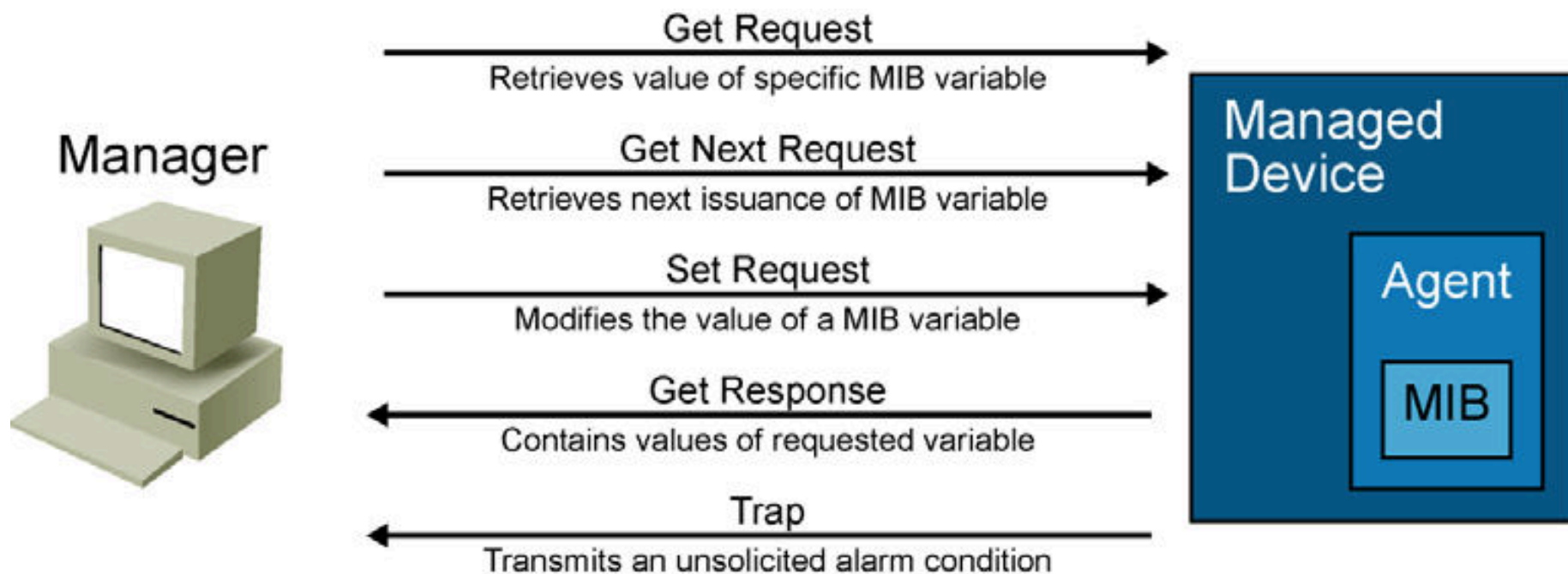
- **SNMP es el protocolo estándar de gestión de red.**
 - SNMP es una solución simple que requiere muy poco código para ponerla en práctica
 - Permite construir fácilmente agentes SNMP para cualquier tipo de dispositivo.
 - SNMP es la base de la arquitectura de gestión de red.
- **Un sistema SNMP consta de dos elementos:**
 - **La aplicación de Gestión de Red (SNMP Manager)**
 - **Los agentes SNMP** (que se ejecutan en un dispositivo administrado)
- **SNMP define como se intercambia la información de gestión entre las aplicaciones de gestión de red y los agentes de administración.**
 - **Los objetos de la Base de Datos (MIB)**, que contienen información en un formato predeterminado que el agente puede utilizar



Introducción a SNMP

- **La aplicación de Gestión de Red (Network Management System, NMS) sondea periódicamente a los agentes SNMP**, que se encuentran instalados en los dispositivos administrados, mediante la consulta de los datos del dispositivo
 - El sondeo SNMP periódico tiene el inconveniente de que hay un retraso entre el momento en que ocurre un evento y el momento en el que se publica en el NMS.
 - Se utilizan para recopilar información del entorno y de rendimiento: uso de CPU, uso de memoria, uso de interfaces,...
- **SNMP utiliza UDP** como mecanismo de transporte IP **para recuperar y enviar la información de gestión**, como las variables MIB
- Los **agentes SNMP**, que residen en los dispositivos gestionados, recogen y almacenan información sobre el dispositivo y su funcionamiento, **responden a las peticiones SNMP y generan “traps” para informar al administrador de determinados eventos**
 - Un “**trap**” puede proporcionar un ahorro sustancial de recursos de red y del agente, eliminando la necesidad de algunas solicitudes de sondeo SNMP
 - Se utilizan para enviar información de eventos en tiempo real
- **El agente recopila los datos y los almacena localmente en la MIB**

Introducción a SNMP



Versiones SNMP

- **SNMP versión 1** (SNMPv1, RFC 1157). Se utilizan **cinco mensajes básicos** SNMP para transferir datos entre el agente y la estación administradora:
 - **Get Request:** Solicita el valor de una variable MIB específica del agente.
 - **Get Next Request:** Usado después de la solicitud inicial, Get Request recupera la siguiente instancia de objeto de una tabla o una lista
 - **Set Request:** Se usa para configurar una variable MIB en el agente.
 - **Get Response:** Utilizada por un agente para responder a Get Request ó Get Next Request solicitado desde el NMS
 - **Trap:** Utilizado por los agentes para transmitir una alarma no solicitada al NMS.
 - Un agente envía un mensaje «trap» cuando se produce una determinada condición: cambio en el estado de un dispositivo, fallo de un componente o del dispositivo o una inicialización o reinicio del agente.

SNMP Versión 2

- SNMPv2 (RFC 1441), pero los miembros del Subcomité IETF no estaban de acuerdo sobre la seguridad y las secciones administrativas de la especificación SNMPv2.
- **SNMPv2 (SNMPv2C, RFC 1901) es la implementación más común de SNMP.**
 - **SNMPv2C despliega el marco administrativo definido en SNMPv1, que utiliza cadenas de comunidad de lectura/escritura para el acceso a los datos.**
- **SNMPv2 introduce dos nuevos tipos de mensajes:**
 - **Get Bulk Request:** Reduce las peticiones y las respuestas repetitivas y mejora el rendimiento al recuperar grandes cantidades de datos (por ejemplo, tablas).
 - **Inform Request:** Alerta NMS de situaciones específicas . En lugar de confirmar los mensajes SNMP Trap, la NMS confirma la solicitud “Inform Request” mediante el envío de un mensaje “Inform Response” al dispositivo que lo solicita.
- **Nota:** Ni SNMPv1 ni SNMPv2 ofrecen características de seguridad. Específicamente, SNMPv1 y v2 **no pueden autenticar el origen de un mensaje de gestión ni proporcionar cifrado**. Debido a la falta de características de seguridad, **muchas implementaciones SNMPv1 y v2 se limitan a una capacidad de sólo lectura**, reduciendo su utilidad a la de un monitor de red.

SNMP Version 3

- SNMPv3 está descrito en los RFCs 3410 hasta el 3415.
 - Agrega **métodos para garantizar la transmisión segura** de datos críticos entre los dispositivos administrados.
 - SNMPv3 presenta **tres niveles de seguridad**:
 - **noAuthNoPriv**: No se requiere ninguna autenticación y no se proporciona ninguna confidencialidad (cifrado).
 - **authNoPriv**: La autenticación está basada en las firmas HMAC-DM5 ó HMAC-SHA. Tampoco se proporciona confidencialidad (cifrado).
 - **authPriv**: autentica el paquete utilizando una firma HMAC basada en MD5 o SHA y cifra el paquete utilizando CBD-DES.
- Los niveles de seguridad implementados para cada modelo de seguridad determinan a que objetos SNMP se puede tener acceso (lectura, escritura, notificaciones).
- Nota: En los routers Cisco, SNMPv3 está implementado en la versión de IOS 12.0 y posteriores.

Recomendaciones SNMP

- Los NMS rara vez necesita permisos de escritura → Configuración en modo ***read-only***
 - Se deben separar las comunidades y credenciales de los sistemas que necesiten acceso de escritura
- Se debe utilizar **vistas**
 - El comando **setup snmp view** permite controlar a qué MIBs y a qué parte de las mismas puede acceder un usuario
 - Por defecto, no hay entradas en la vista SNMP.
 - Funcionan de forma similar a una ACL: Se permite el acceso a la MIB o parte de la MIB explícitamente especificada
- Se debe controlar el acceso mediante ACLs
- Se recomienda SNMPv3 siempre que sea posible
 - Proporciona autenticación, integridad y confidencialidad
- Cuidado con los nombres de comunidad en SNMPv1 y SNMPv2c
 - No deben ser nombres triviales ni por defecto
 - Preferible SNMPv2c (cifra los nombres de comunidad)

Configuración de SNMPv3

1. Configurar listas de acceso SNMP
2. Configurar las vistas SNMPv3 para limitar el acceso a MIBs específicas
3. Configurar grupos de seguridad SNMPv3
4. Configurar usuarios SNMPv3
5. Configurar receptores de *traps* SNMP.
6. Configurar la persistencia de *ifindex* para evitar los cambios del mismo

Configuración de SNMPv3

- Ejemplo:

```
Switch(config)# access-list 99 permit 10.1.1.0 0.0.0.255
Switch(config)# snmp-server view OPS sysUpTime included
Switch(config)# snmp-server view OPS ifDescr included
Switch(config)# snmp-server view OPS ifAdminStatus included
Switch(config)# snmp-server view OPS ifOperStatus included
Switch(config)# snmp-server group groupZ v3 priv
Switch(config)# snmp-server user userZ groupZ v3 auth sha secretpwd2 priv aes
256 secondsecretpwd2
Switch(config)# snmp-server enable traps
Switch(config)# snmp-server host 10.1.1.50 traps version 3 priv userZ cpu
port-security
Switch(config)# snmp-server ifindex persist
```

Verificación de la Configuración

```
SW# show snmp
Chassis: FOC1322V1P5
0 SNMP packets input
...
0 Get-request PDUs
0 Get-next PDUs
0 Set-request PDUs
0 Input queue packet drops (Maximum queue size 1000)
476 SNMP packets output
...
0 Response PDUs
476 Trap PDUs
SNMP global trap: enabled

SNMP logging: enabled
Logging to 10.1.1.50.162, 0/10, 476 sent, 0 dropped.
SNMP agent enabled
```

Verificación de la Configuración

SW# **show snmp view**

```
OPS sysUpTime - included nonvolatile active
OPS ifDescr - included nonvolatile active
OPS ifAdminStatus - included nonvolatile active
OPS ifOperStatus - included nonvolatile active
vldefault iso - included permanent active
vldefault internet - included permanent active
vldefault snmpUsmMIB - excluded permanent active
vldefault snmpVacmMIB - excluded permanent active
vldefault snmpCommunityMIB - excluded permanent active
vldefault ciscoMgmt.252 - excluded permanent active
!...output omitted for brevity
```

SW# **show snmp group**

groupname:	groupZ	security model:v3 priv
readview :	OPS	writeview: OPS
notifyview:	*tv.00000000.00000000.10000000.0	
row status:	active	access-list: 99

Verificación de la Configuración

```
SW# show snmp user
```

```
User name: userZ
```

```
Engine ID: 80000009030000260AC50201
```

```
storage-type: nonvolatile
```

```
active
```

```
Authentication Protocol: SHA
```

```
Privacy Protocol: AES256
```

```
Group-name: groupZ
```